Ansys Fluent on Oracle Cloud HPC

Using Intel Tools

HPC workloads in the Cloud series

Version 0.0 – January 2021

Abstract:

This technical white paper discusses the optimal ways to benchmark Ansys Fluent CFD workloads on Oracle Cloud Infrastructure HPC Instances using Intel technology.

Contents

Overview

As Computer Aided Engineering (CAE) continues to grow, so does the need to run on robust and performant infrastructure. Many customers are looking to move their HPC workloads to the cloud, but this can also be challenging. New cloud users may find that choosing the correct instance can be a daunting task and does not always lead to consistent performance. The compute specifications such as processor and network platform are some of the key influencers of performance, and it is important to consider these when selecting an instance suitable for HPC workloads.

This document should help you overcome some of these concerns by walking through an example setup of a CAE application on Oracle Cloud Infrastructure. Ansys Fluent was chosen as the example because of its popularity for Computational Fluid Dynamics (CFD) simulations which make up a large portion of CAE applications.

Getting Started with Oracle Cloud HPC

For running compute optimized workloads such as Ansys Fluent, Oracle Cloud Infrastructure offers a BM.HPC2.36 instance with Intel Xeon Scalable processors. Oracle's HPC instance is powered by two Intel's Xeon Gold 6154 processors with 18 cores each running at 3.70 GHz. Combined with low latency Remote Direct Memory Access (RDMA) cluster networking, BM.HPC2.36 is one of the top performing instances in the cloud and one of the lowest priced at $2.70 instance / hour. Oracle's HPC Bare Metal solution is suitable for running HPC applications to achieve performance consistent with on-premises environments.

Basic Setup on Oracle Cloud

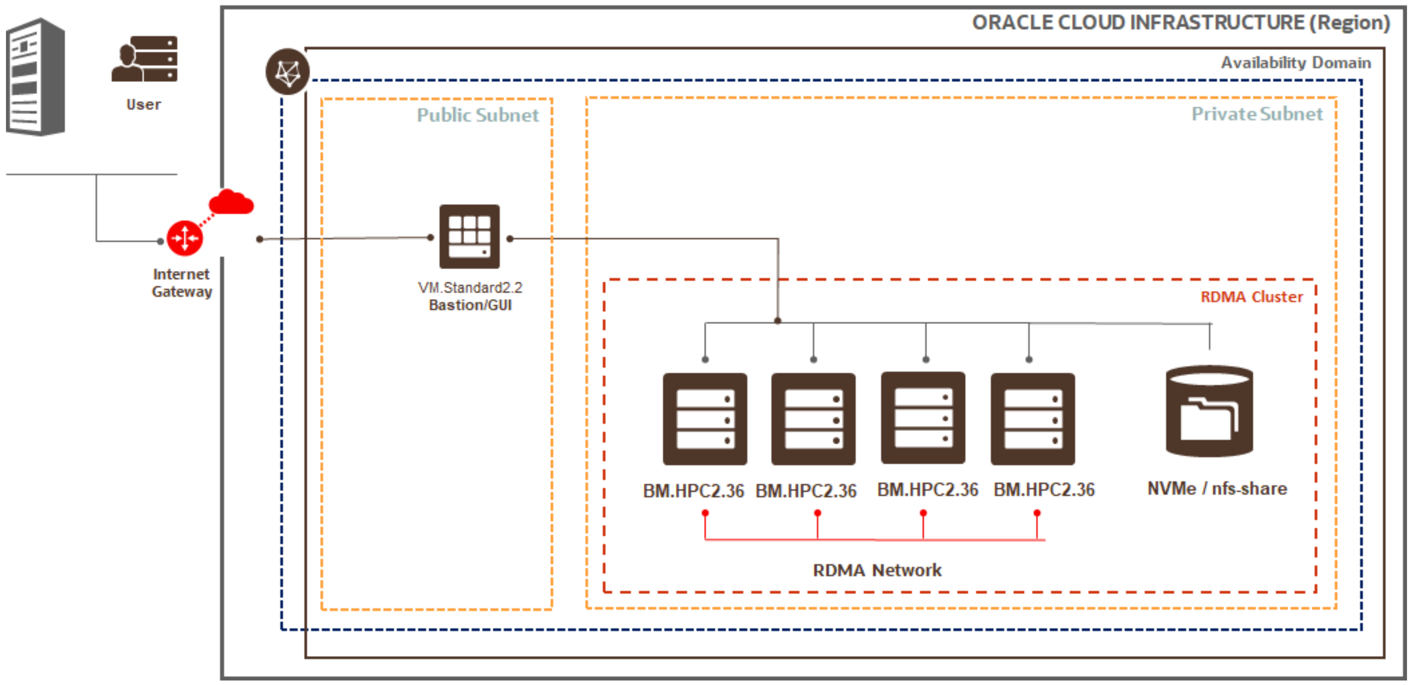
To start using Oracle Cloud, refer to the Getting Started with Oracle Cloud Applications website and signup here. Once you have a tenancy created, refer to Setting up Your Tenancy website to create your compartments and configure Identity Access Management.

Solutions to Launch Oracle Cloud HPC

HPC cluster networks are supported in the following regions, and can be launched through marketplace, via terraform or through the OCI command line interface. In most cases, launching from Marketplace using the HPC Cluster stack is easiest and recommended for beginners. For users that understand terraform and want to modify the infrastructure provisioned or install additional applications on top of it, the OCI HPC CN Terraform scripts are recommended. Both Marketplace and Terraform use the HPC Cluster Networking Image by default to provide you with all the drivers and scripts you need to run a simulation with cluster networking.

Baseline Architecture

If one of the above solutions is used to spin up an OCI HPC cluster, the baseline architecture is, as follows. One virtual cloud network (VCN) is configured with one private subnet and one public subnet. Within the public subnet, a small bastion is provisioned which acts as the connection point from your local machine. The bastion will be accessible through SSH by means of the key used to launch the infrastructure (or by using a GUI if this is enabled). The HPC compute nodes will be provisioned in the private network and only accessible through the bastion.



There are network file system shares to be aware of, including `/mnt/nfs-share/` or `/nfs/scratch`, depending on the HPC image version you are using. The compute servers come pre-installed and configured for RoCEv2 RDMA interconnect. Oracle also provides a default host file `/etc/opt/oci-hpc/hostfile.rdma` which includes all the configured hostnames and uses the low-latency RDMA network across the cluster.

Accessing Oracle Cloud

When it comes to connecting to Oracle Cloud, two of the most common ways to connect are through ssh or through a remote desktop. To connect through ssh keys, Oracle has provided a writeup on the steps.

Alternatively, you could also explore the use of a remote desktop environment which would forward the cloud systems desktop to your local computer. One way to accomplish this is using x2go. It works through SSH and can be setup relatively quickly. To do this, take the following steps:

1. First SSH into the bastion host of your compute cluster.
   ```
   ssh opc@<bastion host public ip> -i /path/to/private_key
   ```

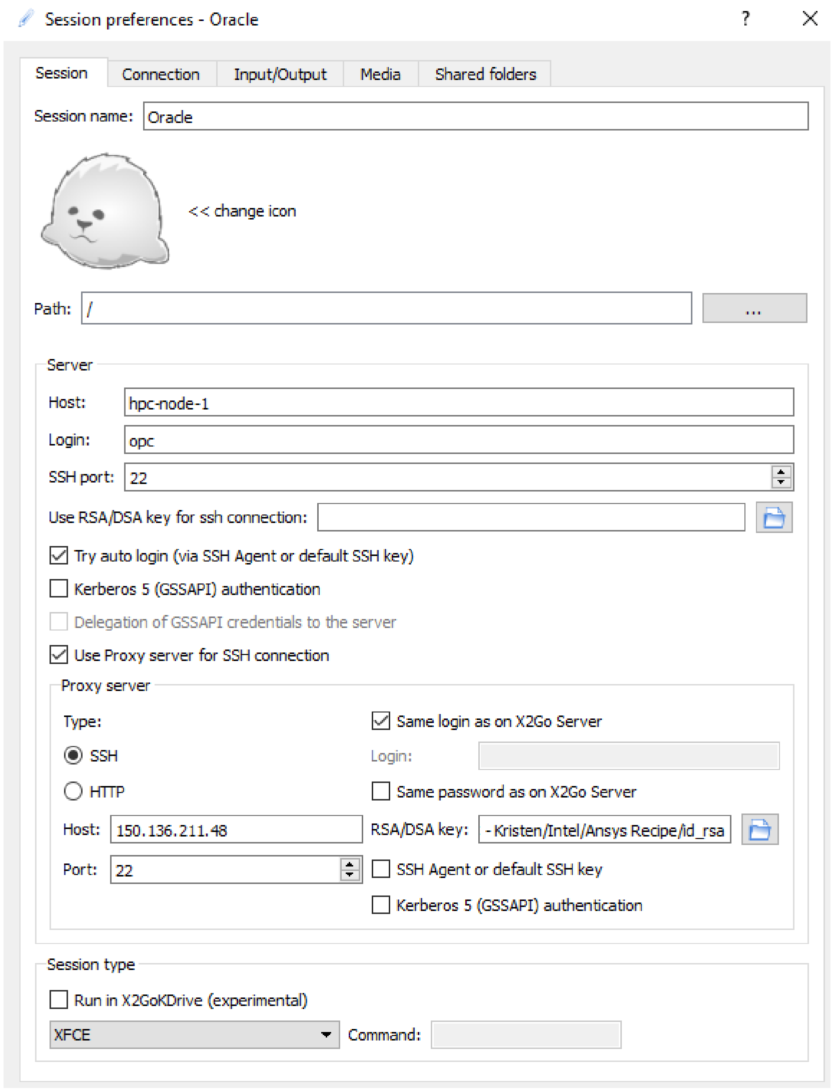
2. Next ssh into the first node (`ssh hpc-node-1`) from the bastion host and add your public key to `~/.ssh/authorized_keys`.

3. Reset the login for the default opc user, if needed:
   ```
   sudo passwd opc
   ```

4. Install the lightweight Xfce desktop environment:
   ```
   sudo yum -y install @Xfce x2goserver x2goserver-fmbindings gvfs-fuse
   ```

5. On your local laptop or workstation you will need to install the x2go client. To configure the client to access the compute nodes, create a new session with the following details:
   - Host: hpc-login-1
   - Login: opc
   - SSH Port: 22
   - Tick "Use RSA/DSA key for ssh connection"
   - Tick “Try auto login (via SSH Agent or default SSH key)”
   - Tick "Use Proxy server for SSH connection"
   - In the zone "Proxy Server" that appeared
     - Type: SSH
     - Host: <the IP of your bastion server>
     - Tick "same login as X2Go Server"
     - Select the file containing the SSH Private key
   - In "Session Type", select "XFCE"

   Now you can click – OK and connect to the new session.



Session name: Oracle

Path: /

Server
Host: hpc-node-1
Login: opc
SSH port: 22
Use RSA/DSA key for ssh connection:
[✓] Try auto login (via SSH Agent or default SSH key)
[ ] Kerberos 5 (GSSAPI) authentication
[ ] Delegation of GSSAPI credentials to the server
[✓] Use Proxy server for SSH connection

Proxy server
Type: [● SSH] ○ HTTP
Host: 150.136.211.48
Port: 22
[✓] Same login as on X2Go Server
Login:
[ ] Same password as on X2Go Server
RSA/DSA key: -Kristen/Intel/Ansys Recipe/id_rsa
[ ] SSH Agent or default SSH key
[ ] Kerberos 5 (GSSAPI) authentication

Session type
[ ] Run in X2GoKDrive (experimental)
XFCE
Command:

Figure 1: x2go client configuration screen

Configure Ansys Fluent

When it comes to running your Fluent workload on Oracle Cloud, you need to consider how best to launch the application. When you run on a cluster that is a shared resource (meaning there are multiple users who are vying for resources), having a workload management system that manages queuing up application runs is highly beneficial. A few other options for running your Fluent application include creating a script to launch and run Fluent or using the command line to run Fluent.

Everyone's use of cloud infrastructure will be different. This paper covers the command line method, as this represents the most basic way of running Fluent. Once you can run via command line, you can create a job script for a workload manager.

Installing Ansys Fluent

Once you can successfully login to your cluster on Oracle Cloud, software installation can now occur. To install Fluent, create a directory `/mnt/nfs-share/upload` and obtain the Ansys Fluent ISO files or the Fluids linux x64 binary from the Ansys Inc website.

1. Define the following linux environmental variables. Replace "www.xxx.yyy.zzz" with your Ansys License Servers IP address.

```
ANS_LICENSE_SERVER="www.xxx.yyy.zzz"
ANS_INSTALL_PATH="/mnt/nfs-share/ansys_inc"
```

2. If you are using an iso file, programmatically mount the Ansys ISO file for installation

```
cd /mnt/nfs-share/upload
typeset instbase="" instopts=""
for iso in ANSYS*_LINX64_Disk?.iso; do
      typeset mntdir="${iso%.iso}"
      typeset mntnum="${mntdir##*Disk}"
      [[ -d "$mntdir" ]] || mkdir –p "$mntdir"
      sudo mount "$iso" "$mntdir"
      [[ "$mntnum" -eq 1 ]] && { instbase="$mntdir"; } || { $instopts+=" -
      media_dir${mntnum} $mntdir"; }
done
```

3. If you are downloading the fluids tarball, use wget.

```
cd /mnt/nfs-share/upload
wget FLUIDS_<version>_LINX64.tar

tar -xf FLUIDS_<version>_LINX64.tar
```

4. Next run the installation for Ansys Fluent using silent install.
   - Iso File Install

```
$instbase/INSTALL -silent -install_dir "$ANS_INSTALL_PATH" -
licserverinfo 2325:$ANS_LICENSE_SERVER:1055 $instopts
```

   - Fluids Tar Install

```
./INSTALL -silent -install_dir "$ANS_INSTALL_PATH" -fluent -
licserverinfo 2325:1055:"$ANS_LICENSE_SERVER"
```

5. Lastly, if you are using the isofile, unmount the previously mounted ISO file using the following command.
```
sudo umount ANSYS*Disk?
```

Installing Models

1. Create a folder in /mnt/nfs-share to hold the models.
```
mkdir /mnt/nfs-share/models
```

2. Download the model(s) in the model folder and untar them.

3. Create a journal file for the model in the models folder. An example for the f1_racecar_140m model is shown below:
```
; Set the logfile
/file/start-transcript /mnt/nfs-share/models/f1_racecar_140m.trn OK

; Enable MKL
(rpsetvar 'amg-coupled/user-defined-smoother "ilu@/mnt/nfs-
share/ansys_inc/v202/fluent/lib/lnamd64/libmklsmoother_sp.so")

; Read dataset
/file read-case /mnt/nfs-
share/models/bench/fluent/v6/f1_racecar_140m/cas_dat/f1_racecar_140m
/solve/initialize/initialize-flow
/file read-data /mnt/nfs-
share/models/bench/fluent/v6/f1_racecar_140m/cas_dat/f1_racecar_140m OK

; pre-run
/solve/set/reporting-interval 5
/solve/iterate 5
/parallel/timer/reset

; Real iterations
/solve/iterate 25

; Checks
/grid check
/parallel/timer/usage
/para part print
/grid mem
/parallel/latency
/parallel/bandwidth

; Write data file
wd fluent_results.dat

; End run
/file/stop-transcript
/exit yes
```

4. Create a hostfile in the models folder:
```
sed 's/$/:36/' /etc/opt/oci-hpc/hostfile.rdma > machinefile
```

Install and use Intel Math Kernel Library (MKL)

Ansys Fluent is distributed with a version of Intel MPI Library. Another Intel optimized library is the Math Kernel Library. This is not distributed by default for versions lower than 20.2. Using Intel MKL can provide a nice speed-up, from 5% - 15%.

1. Download the latest Intel MKL Linux release from Intel's software download center. Copy the package, I_mkl_*.tgz to the cloud headnode system. To make the installation easier we will set two variables:

   ```
   mklvers=”2020.1.217”
   ```
   Where 2020.1.217 represents the version string of the MKL tgz file you just downloaded.

   ```
   ansysdir=”/mnt/nfs-share/ansys_inc/v202”
   ```
   Where this variable is the location of the Ansys Fluent installation.

2. Extract the archive and enter the directory it created:
   ```
   tar -xf l_mkl_${mklvers}.tgz
   cd l_mkl_${mklvers}
   ```

3. Next, we will install Intel MKL by running its installer. Intel MKL includes an end user license agreement, if you have already read and accept its contents you can use the installer flag –accept_eula or read the EULA during the install. (Please note, as of writing the full install takes around 3GB of space).
   ```
   ./install.sh --cli-mode --install_dir "/mnt/nfs-share/intelmkl" --silent
   '--components intel-mkl-core-c__x86_64 --components intel-mkl-gnu-
   c__x86_64 --accept_eula option'

   cd /mnt/nfs-share/intelmkl
   ```

4. Once installed, we need to insure MKL is added to the Fluent third party directory.
   ```
   destdir="${ansysdir}/tp/IntelMKL/${mklvers}/linx64/lib/intel64"
   mkdir --parents "$destdir"
   cp -r compilers_and_libraries_${mklvers}/linux/mkl/lib/intel64_lin/*.so
   "$destdir/"
   ```

5. Edit the path in the Fluent script:
   ```
   sed -i "${ansysdir}"/fluent/fluent*/bin/fluent -Ee "s#IntelMKL/[0-
   9\.]+/#IntelMKL/$mklvers/#"
   ```

6. Ensure that MKL is enabled in fluent by adding this line in the journal file or in the console:
   ```
   (rpsetvar 'amg-coupled/user-defined-smoother
   "ilu@/usr/ansys_inc/v202/fluent/lib/lnamd64/libmklsmoother_sp.so")
   ```

Run Ansys Fluent

Optional baseline benchmarking

Ansys Fluent provides some embedded tests to help identify if there are possible issues with performance prior to launching your application run. The following commands can be run to test performance using Stream-Triad and MPI tests.

These commands configure how Fluent will run and create a default fluentcmd command to run some tests.

```
typeset -a mpinodes=($(cat /etc/opt/oci-hpc/hostfile.rdma))
typeset -i ncpus=$(getconf _NPROCESSORS_ONLN)
typeset fluentcmd="/mnt/nfs-share/ansys_inc/v202/fluent/bin/fluent -g 3d -
mpi=intel -cnf=/etc/opt/oci-hpc/hostfile.rdma"
```

These commands will run different tests through Fluent to check for performance characteristics.

```
$fluentcmd -stream
$fluentcmd -t$(( ${#mpinodes[@]} * ${ncpus:-16} )) -mpitest
$fluentcmd -t$(( ${#mpinodes[@]} * ${ncpus:-16} )) -i <(echo -e \
'/parallel/latency\n/parallel/bandwidth\nexit\n')
```

Ansys Fluent Execution

Ansys Fluent 2020R1 includes Intel MPI 2018 runtime. We suggest launching Fluent with the Intel MPI flags to achieve optimal performance. To start, we will create a couple of variables for node count and journal file.

```
typeset -i nnodes="$(grep -cv ^# /etc/opt/oci-hpc/hostfile)"
typeset journalfile=/mnt/nfs-share/models/f1_racecar.jou
```

Next, the following command will launch Fluent using the total number of cores in the cluster, using IntelMPI and using RoCE v2 RDMA network interfaces.

```
/mnt/nfs-share/ansys_inc/v202/fluent/bin/fluent -g 3d -t$(($nnodes * 36)) -
cnf=machinefile -i $journalfile -platform=intel -mpi=intel -mpiopt="-iface
enp94s0f0 -genv I_MPI_FABRICS shm:dapl -genv DAT_OVERRIDE /etc/dat.conf -genv
I_MPI_DAT_LIBRARY /usr/lib64/libdat2.so -genv I_MPI_DAPL_PROVIDER ofa-v2-cma-
roe-enp94s0f0 -genv I_MPI_FALLBACK 0 -genv I_MPI_FALLBACK=0"
```

There is a lot going on here; lets go into some details about the command line options:

- **-g** – Run without the GUI or graphics
- **3d** – single-precision solver 3D /
- **3ddp** – double-precision solver 3D
- **-t<x>** – amount of process to compute
- **-platform=intel** – AVX2 optimized binary is used which can provide performance improvements
- -**mpi=intel** – specifies that the Intel MPI Library is used for MPI communication
- **-cnf=/etc/opt/oci-hpc/hostfile.rdma** – specifies the path to the hostfile
- **-mpiopt="<options>" –** passes options to IntelMPI Library. Refer to the Cluster Networking Blog for a description of the flags to use based on your mpi version.
- **-cflush** – you can also add -cflush at the end of the command to clear file cache buffers before the run.

The output of the command will yield a .trn file with the total wall clock time for the run. You can calculate the solver rating which is the standard measure of fluent performance:

$$Rating = \frac{\text{24 hours x 3600 seconds}}{\text{total wall clock time}}$$

Ansys Fluent Bench Execution

Note that Ansys Fluent also comes with a fluentbench.pl script that can be used to evaluate performance. To run the fluent script with the necessary Intel MPI flags using DAPL, perform the following:

```
modelname=f1_racecar_140m
N=<number of cores>
```

```
/mnt/nfs-share/ansys_inc/v202/fluent/bin/fluentbench.pl -ssh -noloadchk -
casdat=$modelname -t$N -cnf=machinefile -mpi=intel -mpi=intel -mpiopt="-iface
enp94s0f0 -genv I_MPI_FABRICS shm:dapl -genv DAT_OVERRIDE /etc/dat.conf -genv
I_MPI_DAT_LIBRARY /usr/lib64/libdat2.so -genv I_MPI_DAPL_PROVIDER ofa-v2-cma-
roe-enp94s0f0 -genv I_MPI_FALLBACK 0 -genv I_MPI_FALLBACK=0"
```

Note that instead of using -mpiopt flag, you can also modify each mpirun.fl file in the fluent folder with your own flags, as follows:

Replace:
```
FS_MPIRUN_FLAGS="$FS_MPIRUN_FLAGS -genv I_MPI_ADJUST_REDUCE 2  -genv
I_MPI_ADJUST_ALLREDUCE 2 -genv I_MPI_ADJUST_BCAST 1"
```

To:
```
FS_MPIRUN_FLAGS="$FLUENT_INTEL_MPIRUN_FLAGS -genv I_MPI_ADJUST_REDUCE 2 -genv
I_MPI_ADJUST_ALLREDUCE 2 -genv I_MPI_ADJUST_BCAST 1"
```

And then set the `FS_MPIRUN_FLAGS` to your desired IntelMPI Flags:

```
echo export FLUENT_INTEL_MPIRUN_FLAGS='"-iface enp94s0f0 -genv I_MPI_FABRICS
shm:dapl -genv DAT_OVERRIDE /etc/dat.conf -genv I_MPI_DAT_LIBRARY
/usr/lib64/libdat2.so -genv I_MPI_DAPL_PROVIDER ofa-v2-cma-roe-enp94s0f0 -
genv I_MPI_FALLBACK 0 -genv I_MPI_FALLBACK=0"' | sudo tee -a ~/.bashrc
```

Additional Optimization Suggestions

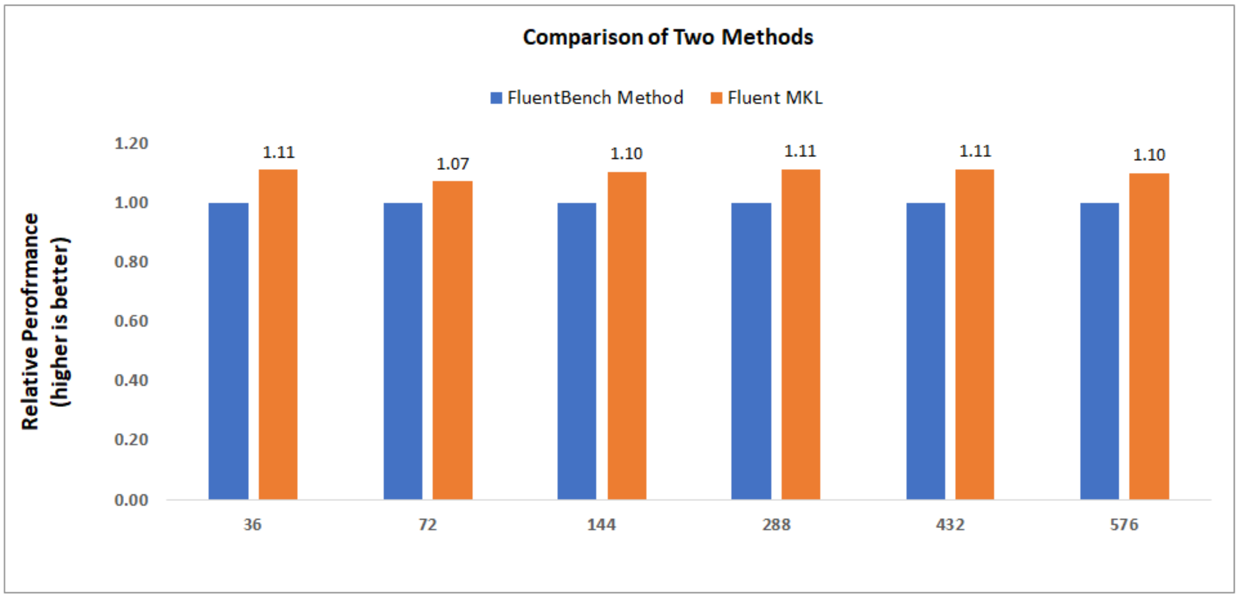
For best performance of Ansys Fluent 2020R2 on Oracle Cloud, it is recommended to use 36 cores per cloud server instead of 72 threads on 1 node. This is because Intel® Hyper-Threading Technology is enabled on Oracle Cloud by default. While Intel® HT Technology provides benefits for some application codes, Fluent performs fastest when using physical cores and not threads. Additionally, by using just 36 cores per server for optimal performance, the license cost is improved since it is based on the number of cores in use, whether physical or threads. You can disable hyperthreading on the cluster or else pin the processes by adding the following to the -mpiopt flag:

```
-genv I_MPI_PIN_PROCESSOR_LIST=0-35 -genv I_MPI_PROCESSOR_EXCLUDE_LIST=36-71
```

Optimization Benefits

Optimizations discussed in this paper will allow users to take advantage of Intel features and software enhancements, without needing to recompile anything. At the core, using Intel MLK, IntelMPI and the correct MPI Flags provides better performance and speed up of various model sizes.

To illustrate some of these options in action, the performance using the Ansys Fluent MKL approach was compared with the standard Ansys Fluent fluentbench script for f1_racecar_140m. On average, the use of MKL and Intel options provided approximately a 10% increase in performance from 36 to 576 cores.



More information

To find Ansys Fluent standard benchmarks, you must be a customer or Ansys Partner. Sign in to the Ansys Customer Portal to download each model. Read about the model cases and how to install them via the Benchmarks_Description.pdf and README_INSTALL.